БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

КИБЕРНЕТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ · Том **3**, № 2
CYBERNETICS AND INFORMATION TECHNOLOGIES · Volume **3**, No 2

София . 2003 . Sofia

Model Reference Adaptive Neural Control of a Variable Structure System

Moises Bonilla, Ieroham Baruch*, Jose Martin Flores*, Maria M. Goire*, Boyka Nenkova***

** Department of Automatic Control, CINVESTAV-IPN, 07360 Mexico D.F., MEXICO*
*** Institute of Information Technologies, 1113 Sofia,*

Abstract: *The aim of this paper is to propose a model reference adaptive neural control of a variable structure plant, described by an implicit realization with variable order and parameters, using only output feedback. The neural control scheme proposed is composed by two recurrent neural networks, named: neuro-identifier and neuro-controller. A variable structure plant model together with the realized adaptive neural control are simulated by means of the MatLab-Simulink and the obtained simulation results are compared with those obtained by the use of an ideal implicit control, applying the true descriptor variable. The simulation results show a great similarity of the obtained graphics for both control schemes, which demonstrated the applicability of the proposed adaptive neural control.*

Keywords: *Model reference adaptive control, variable structure plant, recurrent neural networks, backpropagation learning.*

1. Introduction

The application of Neural Network (NN) modeling to system identification, prediction and control was discussed by many authors, [12, 13, 14]. Mainly, two types of NN models are used: Feed forward (FFNN) and Recurrent (RNN). Narendra and Parthasarathy [14], applied FFNN for system identification and direct model reference adaptive control of various non-linear plants. They considered four plant models with a given structure and supposed that the order of the plant dynamics is known. H u n t et al. [12], J i n et al. [13], surveyed some schemes of NN and RNN applications to control, especially for direct model reference adaptive control. All drawbacks of the described in the literature NN models could be summarized as follows: 1) there exists a great variety of NN models and universality is missing [12, 14]; 2) all NN models are sequential in nature as implemented for systems identification (the FFNN model uses one or two tap-delays in the input [14] and RNN models usually are based on the autoregressive model [12], which is one-layer sequential one); 3) some of the applied RNN models are not trainable, others are not trainable in the feedback part, [12]; 4) most of them are dedicated to a SISO and not to a MIMO applications [14]; 5) in more of the cases, the stability of the RNN is not considered [12], especially during the learning; 6) in the case of FFNN application for systems identification, the plant is given in one of the four described by N a r e n d r a and P a r t h a s a r a t h y [14] plant models, the linear part of the plant model, especially the system order, has to be known and the FFNN approximates only the non-linear part of the plant model, [7, 14] all these NN models are nonparametric ones, [12] and all NN models do not perform state and parameter estimation at the same time [1].

B a r u c h et al. [1, 2] in their previous paper, applied the state-space approach to describe RNN in an universal way, defining a Jordan canonical two-layer RNN model, named Recurrent Trainable Neural Network (RTNN). This NN model is a parametric one, permitting the use of the parameters for control systems design, obtained during the learning [1, 2]. Furthermore, B a r u c h et al. [2] used the RTNN model as a system state predictor/estimator and parameter identifier in an indirect adaptive control system of nonlinear plants. The aim of this paper is to use the RTNN as an identification and control tool in a direct model reference neural control system of a variable structure plant and to compare the proposed neural control with the ideal implicit one. The proposed model reference adaptive neural control system is studied by means of a continuous-time variable structure dynamic plant model and simulation results, compared with the implicit control, are given.

2. Model reference adaptive neural control

This part of the paper is related to a model reference adaptive control scheme whose aim is to track a reference obtained as the output of a given reference model. This purpose is realized using the neural network approach that is able to generate identification and control models without the necessity to define the parametric structure of the system and to adjust so as to fit the acquired system data, [1]. First, the work hypothesis is stated. Next, an adequate scheme of a neural identifier is defined. Next, the topology of the neural network is described. Then the training algorithm is shown. And finally, the neural controller is designed.

2.1. Assumption

Given a variable structure system which changed its internal structure between switching intervals, it is going to be assumed that in each interval the system can be modeled as:

$$x_p(k+1) = \psi\,[x_p(k),\, u(k)],$$
$$y_p(k) = \varphi\,[x_p(k)], \tag{1}$$

where u is the input, x_p is the internal state and y is the output of dimensions m, n_p, l, respectively.

2.2. Neural identifier (Forward modelling)

The procedure of training a neural network to represent the forward dynamics of a system will be named forward modelling, [12, 13]. The neural model is placed in parallel with the system and the error between the system and the network output is used as a training signal. This learning structure is a classical supervised learning problem where the teacher (in this case the system output) provides the target values directly in the output coordinate system of the learner (the neural model).

2.3. Recurrent Trainable Neural Network topology

A discrete time model of Recurrent Trainable Neural network (RTNN) is proposed with dynamic backpropagation (BP) weight updating rule, taken from the papers of B a r u c h et al. [1, 2]. The RTNN structure is described by the following equations:

$$x_n(k + 1) = J\, x_n(k) + B\, u_n(k), \tag{2}$$

$$z(k) = S\, [x_n(k)], \tag{3}$$

$$y(k) = S\, [C\, z(k)], \tag{4}$$

$$|\, J \,| < 1, \tag{5}$$

where $x_n(\cdot)$ is an n-internal state vector: $u_n(\cdot)$ is an m-input vector; $y_n(\cdot)$ is an l-output vector, $z(\cdot)$ is an auxiliary variable; $S(\cdot)$ is a vector-value sigmoid function with elements

$$S_i(.) = \frac{1}{1 + e^{-(.)}}; \tag{6}$$

J is a $n{\times}n$ weight state block-diagonal matrix with $l{\times}l$ block structure; B, C are $n{\times}m$ and $l{\times}n$ weight input and output matrices with structure, corresponding to the structure of J. The condition (5) is a RTNN model stabilizing condition. As it can be seen from equations (2) to (4), the given RTNN model is a two-layer hybrid one, with one feed forward output layer and one recurrent hidden layer. It is also completely parallel and parametric one, so it is useful for identification and control purposes.

From (2), (3), (4) it is clear that the structure of the RTNN is similar to the structure of the plant so the RTNN could be trained to emulate $\psi(\cdot)$ and $\varphi(\cdot)$. In this way the resulting neural identifier, given by (2), (3) and (4) can be described by:

$$x_n(k + 1) = \psi\, [x_n(k), u_n(k)], \tag{7}$$

$$\hat{y}_n(k+1) = \varphi\, [x_n(k+1)], \tag{8}$$

where $\hat{y}_n$ is the estimated plant output, which is the output of the neural identifier and x_n is the internal state vector of the RTNN. Here the plant and the RTNN states, x_p and x_n are independent variables.

Some RTNN characteristics: the RTNN application to process control requires defining of the following characteristics and properties of the RTNN model:

• **Parallel distributed processing:** The architecture proposed leads immediately to parallel implementation.

• **Multivariable systems:** The RTNN naturally process many inputs and have many outputs; they are readily applicable to multivariable systems.

• **Nonlinear systems:** The RTNN call great promises in the field of nonlinear control problems solutions. So it is clear that the RTNN has the ability to approximate nonlinear systems.

• **RTNN training:** In the same way as in the case of FFNN, here the learning BP through time algorithm for RTNN, could be derived using the sensitivity model [14]. The most general BP learning algorithm is:

(9) $$W_{ij}(k + 1) = W_{ij}(k) + \eta\, \Delta W_{ij}(k) + \alpha\, \Delta W_{ij}(k - 1),$$

where W_{ij} is the (i, j)-th weight element of each weight matrix C, J, B of the RTNN model to be updated; ΔW_{ij} (ΔC_{ij}, ΔJ_{ij} and ΔB_{ij}) is the weight correction of W_{ij} (C, J and B); η, α are learning rate parameters, k is the iteration number.

The ΔC_{ij}, ΔJ_{ij} and ΔB_{ij} updates of the model weights C, J, B are given by:

(10) $$\Delta C_{ij} = e_j(k)\ y_j\,[1 - y_i(k)]\ z_j(k),$$

(11) $$R_i = C(k)\ e(k)\ z_j\ (k)[1 - z_i(k)],$$

(12) $$\Delta\, J_i\ (k) = R_j\, x_j\,(k - 1),$$

(13) $$\Delta\, B_{ij}\ (k) = R_j\, u_j\,(k),$$

where the J_i weights are restricted as it is indicated in (5), in order to assure the RTNN stability during learning. Using this algorithm, the training of the neuro identifier could be easily accomplished via on-line learning scheme in order to minimize a norm of the instantaneous identification error. The identification error e_M and the performance index E_M to be minimized are:

(14) $$e_{id}\ (k) = y_p(k) - \hat{y}_n(k),$$

(15) $$E_{id} = (1/2)\ e^2_{id}\ (k).$$

2.5. Neural controller

The neural controller is another RTNN, which means that the controller is a dynamic system, given in the form:

$$x_c(k + 1) = \psi\,[x_x(k), u_n(k)],$$

$$u_p(k) = \varphi\,[x_c(k)].$$

The controller input here is the vector $v(k)$, defined as:

(16) $$v_n(k) = [\hat{y}_n(k{+}1);\ e_c(k{+}1)],$$

where $\hat{y}_n(k{+}1)$ is the estimate output prediction, given by the identifier, and $e_c(k{+}1)$ is the error prediction. The neural controller is trained in order to minimize the instantaneous control error performance index:

(17) $$E_c = (1/2)\ e^2_c\ (k),$$

(18) $$e_c\ (k) = y_{\mathrm{ref}}\ (k) - y_n(k).$$

In the above equation $y_{\mathrm{ref}}\ (k)$ is the output of the reference model. In the model reference adaptive control the reference is acquired using a reference model, given in linear form. The complete block diagram of the neural control system, containing two RTNN is shown in Fig. 1.

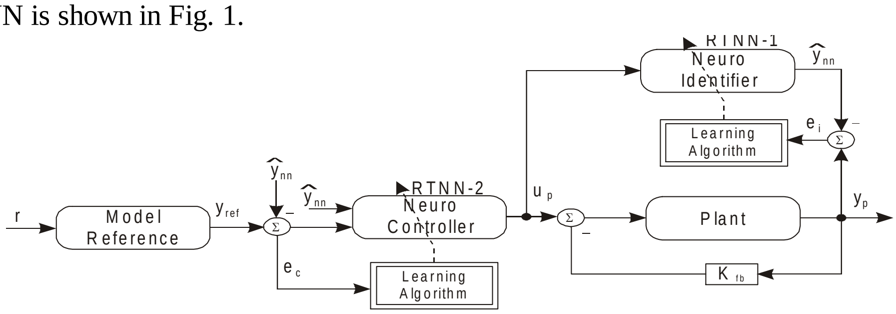

Fig. 1. Block-diagram of the proposed neural model reference adaptive control scheme

This identification and control scheme is proved and compared with the well known implicit control by means of a simulation example.

3. An implicit description of a variable structure control system. Illustrative example

Rosenbrock [15] was the first to introduce the Implicit Descriptions,

$$E\dot{x}(t) = Ax(t) + Bu(t);\ y(t) = Cx(t), \tag{19}$$

where E: $X \to \underline{X}$, A: $U \to \underline{X}$, B: $U \to \underline{X}$ and C: $X \to Y$, are linear operators of appropriate dimensions, as a generalization of the standard State Space case ($E = I$). In [6] it was shown that when $\dim \underline{X} \le \dim X$, it is also possible to describe linear systems with an internal Variable Structure. Indeed, when and if the system is solvable (i.e. possesses at least one solution), the solutions are generally non unique. In some aspect, there is a degree of freedom in (19), which can be used, for instance, to take into account, a possible *structure variation* in an implicit way. Different kinds of structure variations have beentaken into account in the papers of B a s e r and S c h u m a c h e r [3], and H e e l m s et al. [11].

Let us consider as an example (this example will be used further as a simulation example) the following implicit flat description:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\dot{x}(t) = \begin{bmatrix} -1 & -1 & 1 \\ 0 & -1 & 0 \end{bmatrix}x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u(t); \quad y(t) = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}x(t), \tag{20}$$

with the additional linear constraint $[\ a\ \ b\ \ c\ ]x(t) = 0$.

• If $[\ a\ \ b\ \ c\ ] = [\ 0\ \ -1\ \ \ 1\ ]$, namely $x_2 = x_3$, the system behaves as a first order one:

$$\begin{bmatrix} \dot{\bar{x}}_1(t) \\ \dot{\bar{x}}_3(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u(t); \quad y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} \bar{x}_1(t) & \bar{x}_3(t) \end{bmatrix}^T, \tag{21}$$

($\bar{x}_1 = x_1$ and $\bar{x}_3 = x_1 + x_3$) with input-output description $\dot{y}(t) + y(t) = u(t)$.

• If $[\ a\ \ b\ \ c\ ] = [\ 0\ \ \ 1\ \ \ 0\ ]$, namely $x_2 = 0$, the system behaves as of second order:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u(t); \quad y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1(t) & x_3(t) \end{bmatrix}^T, \tag{22}$$

with input-output description $\ddot{y}(t) + \dot{y}(t) = u(t)$.

• Finally, if $[\ a\ \ b\ \ c\ ] = [\ 0\ \ -1\ \ \ 5\ ]$, namely $x_2 = 5x_3$, the system behaves as a second order one with a dominant zero:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} -1 & -4 \\ 0 & -5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)\ ;$$
(23)
$$y(t) = \begin{bmatrix} 1 & 5 \end{bmatrix} \begin{bmatrix} x_1(t) & x_3(t) \end{bmatrix}^{\mathrm{T}},$$

with input-output description $\ddot{y}(t) + 6\dot{y}(t) + 5y(t) = 5\dot{u}(t) + u(t)$.

In the general case, if the matrices E and A are not square, F r a n k o w s k a [9], extended the geometric characterization of the controllable subspace $R_x{}^*$, using differential inclusion techniques, as follows:

$$R_x{}^* = V_x{}^* \cap S_x{}^*,$$

where $V_x{}^* = \sup\ \{T \subset X: AT = ET + \mathrm{Im}\ B\}$ and $S_x{}^* = \inf\ \{T \subset X: T = E^{-1}(AT + \mathrm{Im}\ B)\}$.

In the case of non square (flat) E, A matrices, one may be faced to controllable systems, even in the absence of any input. This is possible because of the existence of the free descriptor variables (degree of freedom), acting as internal controls. Indeed, for the autonomous system $[\ 1\ \ \ 0]\ \dot{x}(t) = [\ 0\ \ \ 1]\ x(t) + [0]\ u(t)$, we have $R_x{}^* = X$; since $AX = EX \subset EX + \mathrm{Im}\ B$ implies that $V_x{}^* = X$, and $E^{-1}(A\{0\} + \mathrm{Im}\ B) = \mathrm{Ker}\ E \neq \{0\}$, $E^{-1}(A\mathrm{Ker}\ E + \mathrm{Im}\ B) = E^{-1}\underline{X} = X \neq \mathrm{Ker}\ E$, $E^{-1}(X + \mathrm{Im}\ B) = E^{-1}A\ X = E^{-1}E\ X = X$, imply that $S_x{}^* = X$. In order to avoid such pathologies, the concept of *output dynamics* assignment has been introduced by B o n i l l a et al. [6], which guarantees controllability by means of the control input, based on proportional and derivative feedbacks.

Definition 1 (B o n i l l a et al. [6]). The implicit system (19) is called *reachable with output dynamics* assignment if it is reachable and if the supremal observable part of the spectrum of $\lambda\ (E - BF_d) - (A + BF_p)$ can be chosen arbitrarily with the help of the control law $u(t) = F_d\,\dot{x}(t) + F_p x(t)$.

T h e o r e m 1 (Bonilla et al. [6]). The implicit system is reachable with output dynamics assignment if and only if the following two geometric conditions hold:

$$R_x{}^* = X$$

and $\dim\ (V_x{}^* \cap \mathrm{Ker}\ E) - \dim\ (\mathrm{Im}\ B | (\mathrm{Im}\ B \cap \mathrm{Im}\ E)) \leq \ldots \leq \dim\ (\ V^* \cap E^{-1}\mathrm{Im}\ B)$, where $V^* = \sup\ \{T \subset \mathrm{Ker}\ C: AT = ET + \mathrm{Im}\ B\}$.

Continuing with the illustrative example, it is shown by B o n i l l a et al. [5], that (21) is reachable with output dynamics assignment and applying the methodology given by B o n i l l a et al. [6] to synthesize a linear controller for such an implicit flat description, the following control law is obtained:

$$u(t) = \begin{bmatrix} -\dfrac{1}{\tau_0} & (1 - \dfrac{1}{\tau_0}) & 0 \end{bmatrix} x(t) + \ldots$$
(24)
$$\ldots + \begin{bmatrix} -1 & -1 & 1 \end{bmatrix} \dot{x}(t) + \frac{1}{\tau_0} R(t).$$

After the change of variables: $\zeta_1(t) = x_1(t)$, $\zeta_2(t) = x_3(t)$, $\zeta_3(t) = x_1(t) + x_2(t)$, applying the control law (24), the flat description (20) takes the following form:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\zeta}(t) = \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -1/\tau_0 \end{bmatrix} \zeta(t) + \begin{bmatrix} 0 \\ -1/\tau_0 \end{bmatrix} R(t) ,$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \zeta(t) .$$

Thus, the degree of freedom has been made unobservable, in other words, the variation of the structure is no longer presented at the output. The closed loop system behaves as

$$\tau_0 \dot{y}(t) + y(t) = R(t) ,$$

whether the constraint:

$$\begin{bmatrix} a & b & c \end{bmatrix} \in \left\{ \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}, \ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \ \begin{bmatrix} 0 & -1 & 5 \end{bmatrix} \right\}$$

is active. In the paper of B o n i l l a et al. [5], a procedure for approximating the non-proper control law (24), guaranteeing internal stability, by the following proper controller (see also [4, 7]), is proposed. The control law obtained is the following:

$$u(t) = \begin{bmatrix} -\dfrac{1}{\varepsilon_c} & \left(1 - \dfrac{1}{\varepsilon_c}\right) & -\left(\dfrac{1}{\tau_0} - \dfrac{1}{\varepsilon_c}\right) \end{bmatrix} x(t) - \ldots$$

(25)
$$\ldots - \left(\frac{1}{\tau_0} - \frac{1}{\varepsilon_c}\right) x_4(t) + \frac{1}{\tau_0} R(t),$$

(26)
$$\varepsilon_c \dot{x}_4(t) + x_4(t) = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix} x(t) .$$

Applying the control law (25)–(26) to the plant (20), the following closed loop system, is obtained:

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{matrix} 0 \\ 1 \end{matrix} \\ \begin{matrix} 0 & -\varepsilon_c & \varepsilon_c \end{matrix} & \varepsilon_c \end{bmatrix} \underline{\zeta}(t) = \ldots$$

$$\ldots = \begin{bmatrix} \begin{bmatrix} 0 & 1 & -1 \\ 0 & 0 & -1/\tau_0 \end{bmatrix} & \begin{matrix} 0 \\ -1/\tau_0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & -1 \end{bmatrix} \underline{\zeta}(t) + \begin{bmatrix} \begin{bmatrix} 0 \\ 1/\tau_0 \end{bmatrix} \\ 0 \end{bmatrix} R(t),$$

$$y(t) = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & 0 \end{bmatrix} \underline{\zeta}(t),$$

where

$$\underline{\zeta}_1(t) = x_1(t), \qquad \underline{\zeta}_3(t) = x_1(t) + x_2(t),$$

$$\underline{\zeta}_2(t) = x_3(t), \quad \underline{\zeta}_4 = -x_1(t) - x_2(t) + x_3(t) + x_4(t)$$

Since the control law (the non-proper controller (24) and the proper (25), (26) requires knowledge of the descriptor variable, it is necessary to estimate it. Since the synthesis of a *descriptor variable* deeply depends on the knowledge which *internal* *structure* is active, a structure detector has been proposed by B o n i l l a et al. [8], based on a normalized gradient adaptation algorithm, projected along a given hypersphere, with the aim to identify in finite time which *internal structure* is presented. As another alternative, a neural classifier has been proposed by G o i r e et al. [10] in order to estimate the descriptor variable.

The next chapter of this paper applies a model reference control scheme for variable structure systems using output feedback in order to avoid the descriptor variable reconstruction. A variable structure plant model tests the proposed model reference neural control system and the obtained simulation results are compared with the results obtained by the use of the classical implicit control.

4. Simulation results

We show here some MatLab-Simulink simulation results of the variable structure system (20), controlled by the neuro-controller, described in Section 2, and by the implicit control (25), (26). During the simulation, the variable structure system switches among systems (21), (22) and (23), every 150 s. The simulation results assume that the reference model is a first order system described by:

$$Y_{\mathrm{ref}}/R = 0.2/(s + 0.2).$$

The parameters of the neural networks, used in the simulations, are shown in Table 1.

Table 1. Neural networks parameters

Parameter	Value
Internal nodes, n	4
Momentum gain, α	0.00015
Training gain, η	0.00015
Sampling time, T_s	0.01sec

The reference $R(t)$ is a symmetric square wave with a period of 100 s and amplitude of 1. The gain K_c=10 is used to amplify the effect of e_c in weight adjustment for the neuro controller. The response of the reference model output and the variable structure system output are shown in Fig. 2. Here it can be seen that the behaviour of the controlled system is close to the desired model reference output. Note that there are overshoots when the system switch occurs, because of changes in the initial conditions.

Fig. 3 shows the control signal, generated by neural network and the control error between the variable structure system output and the reference model.

Figs. 4 and 5 show the simulation results for variable structure system with the implicit control using the true descriptor variables of the system.

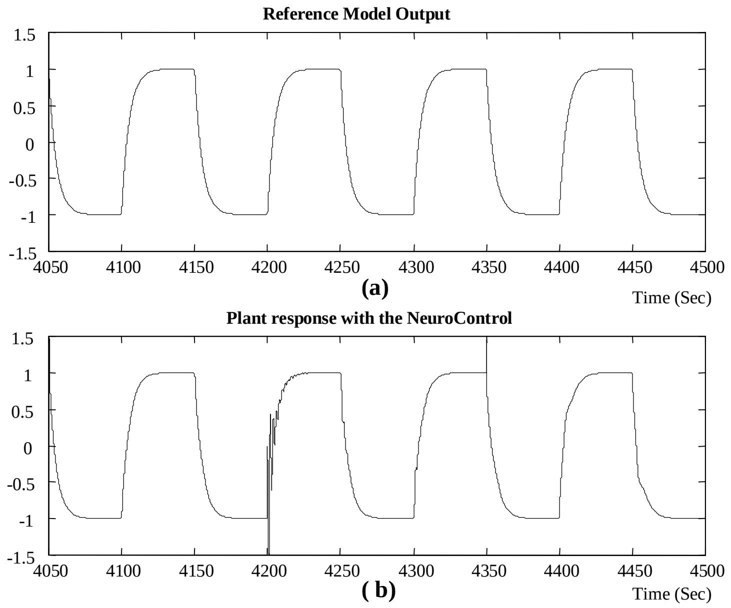

Fig. 2. Model Reference Adaptive Control simulation results – reference and system output signals: a – reference model output; b – plant output with neurocontrol (the time given in seconds)

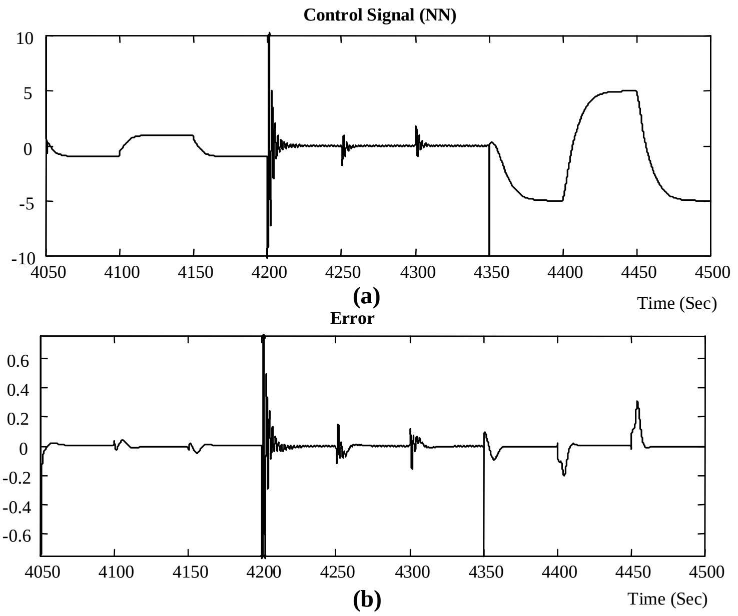

Fig. 3. Model Reference Adaptive Control simulation results – control and system error signals: a – neurocontrol signal; b – systems error ($e_c = y_{\mathrm{ref}} - y_p$) (the time given in seconds)

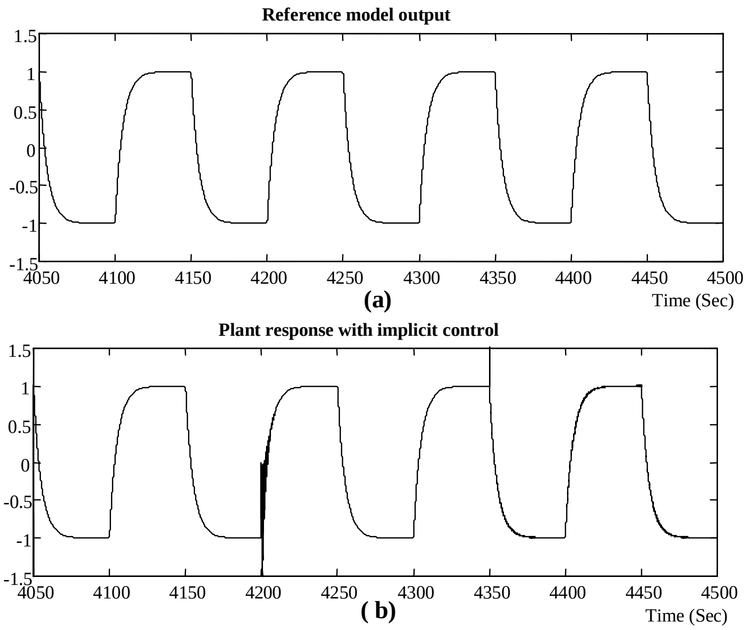

Fig. 4. Implicit Control simulation results – reference and system output signals: a – reference model output; b – plant response using the implicit control, given by the eqns. (25), (26) (the time in seconds)

The implicit control signal, computed using the equations (25), (26), and the control error are shown in Fig. 5.

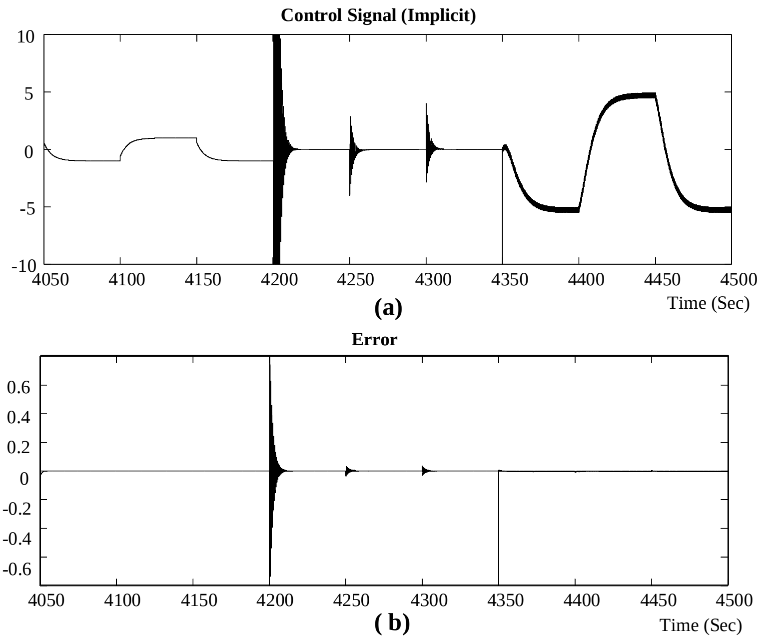

Fig. 5. Implicit Control simulation results – control and error signals: a – implicit control signal; b – systems error ($e_c = y_{\mathrm{ref}} - y_p$)(the time given in seconds)

It is interesting to note that the output and control behaviour using the implicit control and the one, when using the proposed neuro controller, are very similar.

5. Conclusions

The aim of this paper is to propose a model reference control scheme for variable structure systems, described by the implicit realization (19), using only output feedback, in order to avoid the descriptor variable reconstruction. A neural controller scheme is proposed as an alternative solution. This neural controller scheme is composed by two stages, namely: neuro identifier and neuro controller (see Fig. 1). The aim of the neuro identifier is to predict the output behavior one step ahead. A learning algorithm, using the predicted output error, adjusts the parameters of the neuro identifier. The neuro controller is composed by a dynamic neural network whose inputs are the predicted error between the model reference and the output estimation, and the predicted output estimation itself. The parameters of the neuro controller are adjusted by the same learning algorithm, using the error between the model reference and the output estimation. The model reference control scheme is based on systems on-line identification and control. Notice that the proposed control strategy does not require full knowledge of which internal structure is active, i. e. whether the system is of first order or second order, or second order with dominant zero. The recurrent neural network used has constraints in its feedback weights, in order to guarantee stability during the learning. The neural network uses the input-output signals for training, and the applied learning law allows on-line identification and control. The system performance, obtained by the neural approach, is close to that, given by the implicit control law (25), (26), using the descriptor variables of the system.

References

1. Baruch, I., I. Stoyanov, T. Arsenov. An improved RTNN model for dynamic systems identification and time-series prediction. – In: S. Domek, Z.Emirsajlov, R.Kaszynski (Eds.). Proceedings of the 4th International Symposium on Methods and Models in Automation and Robotics, 2 , (Poland: TU of Szczecin), 26-29, Aug. , 1997, Miedzyzdzoje, Poland, 711-717.
2. Baruch, I, J. M. Flores, R. Garrido, E. Gortcheva. An indirect adaptive neural control of nonlinear plants. Proceedings of the International Joint Conference on Neural Networks, 24-27 July, 2000, Como, Italy, 66-103.
3. Baser, U., J. M. Schumacher. Descriptor representations of jump behaviors. – In: Proceedings of the IFAC Conference on System Structure and Control, 8-10 July, 1998, Nantes, France, 159-164.
4. Bonilla, M., M. Malabre. Variable structure systems via implicit descriptions. – In: Proceedings of the 1st European Control Conference, **1**, 2-5 July, 1991, Grenoble, France, 403-408.
5. Bonilla, M., M. Fonseca, M. Malabre. On the implementation of some variable structure systems control laws (the case of two points variating systems). – In: Proceedings of the 32nd Conference on Decision and Control, 1993, 895-896.
6. Bonilla, M., G. Lebret, M. Malabre. Output dynamics assignment for implicit descriptions. – Circuits, Systems and Signal Processing, Special Issue on Implicit Robust Systems, **13**, 1994, 349-359.
7. Bonilla, M., M. Malabre, M. Fonseca. On approximation of non proper control laws. – International Journal of Control, **68**, 1997, 775-796.
8. Bonilla, M., M. Goire, S. Mondie. Adaptive structure detector. – In: Proceedings of the American Control Conference, 2000, 2179-2183.
9. Frankowska, H. On controllability and observability of implicit systems. – Systems and Control Letters, **14**, 1990, 219-225.
10. Goire, M. M., M. Bonilla, S. Mondie. Neural network reconstructor for implicit systems. – In: Proceedings of the American Control Conference, 2000, 3437-3438.

11. H e e m e l s, W.P.M.H., J. M. S c h u m a c h e r, S. W e i l a n d. Linear complementarity systems. – SIAM Journal on Applied Mathematics, **60**, 2000, 1234-1269.
12. H u n t, K. J, D. S b a r b a r o, R. Z b i k o w s k i, P. J. G a w t h r o p. Neural networks for control systems, A survey. – Automatica, **28**, 1992, 1083-1112.
13. J i n, L., P. N. N i k i f o r u k, M. M. G u p t a. Model reference control of discrete-time nonlinear systems using neural networks. – Control Theory and Advanced Technology, **10**, 1995, 1379-1399.
14. N a r e n d r a, K.S., K. P a r t h a s a r a t h y. Identification and control of dynamical systems using neural networks. – In: IEEE Transactions on Neural Networks, **1**, 1990, 4-27.
15. R o s e n b r o c k, H.H. State Space and Multivariable Theory. New York, Wiley, 1970.

Адаптивно невронно управление с еталонен модел на системи с променлива структура

Мойсес Бония, Йерохам Барух*, Хосе Мартин Флорес*, Мария М. Гойре*, Бойка Ненкова***

** Department of Automatic Control, CINVESTAV-IPN, 07360 Mexico D.F., MEXICO*

*** Институт по информационни технологии, 1113 София*

(Р е з ю м е)

Цел на настоящата статия е да предложи адаптивно невронно управление с еталонен модел на обект с променлива структура, описан чрез имплицитна реализация с променлива размерност и параметри, с използване само на обратна връзка по изхода. Предложената невронна схема на управление е съставена от две рекурентни невронни мрежи, а именно: невронен идентификатор и невронен контролер. Моделът на обекта с променлива структура, заедно с реализираната схема за невронно управление, са симулирани с използване на Матлаб Симулинк и получените графични резултати са сравнени с тези, получени с използване на идеално имплицитно управление, използващо истинската дескрипторна променлива. Симулационните резултати показват голямо сходство на получените графики чрез двете схеми на управление, което демонстрира приложимостта на предложеното адаптивно невронно управление.